A Compositional View of Deep Networks as Fractional Associative Memories

Motivation

Neural networks are often described as learning hierarchical feature representations.

Another possible viewpoint is that they learn **hierarchies of fractional associative memories**.

This note develops that intuition. It is intended as a conceptual framework that may help motivate a later mathematical treatment rather than as a formal theory.

From associative memories to fractional associative memories

Classical associative memory is usually described as storing complete associations.

Given an input cue, the memory retrieves the corresponding stored response.

Deep neural networks appear to operate differently.

- No single neuron appears to represent a complete concept.
- No single layer appears to perform the entire computation.
- Knowledge is distributed across many parameters and many layers.

This suggests a different picture.

Rather than storing complete memories, each computational fragment stores only a **small reusable statistical component** of the overall computation.

These are referred to here as **fractional associative memories**.

What does a fractional associative memory store?

A natural first assumption is that each memory stores examples.

A more interesting possibility is that it stores something else entirely.

Instead, each fractional associative memory may store:

> **The average linear transformation that has historically been most useful within a particular context.**

It is therefore not remembering individual training examples.

Instead it is remembering the statistical behaviour of a neighbourhood of the data manifold.

Local statistical models

Imagine the training data lying on a complicated manifold.

Each routing context repeatedly receives examples drawn from a small region of that manifold.

During learning, the parameters associated with that context gradually become better at modelling that region.

Over time they learn something like:

- What transformations usually work here?
- What statistical relationships usually hold here?
- How should representations typically evolve from this neighbourhood?

The stored knowledge therefore becomes a local statistical summary rather than a collection of memorized examples.

Why linear mappings?

Linear mappings have important advantages.

- They are simple.
- They are stable.
- They are easy to optimize.
- They are often excellent approximations over sufficiently small neighbourhoods.

Instead of approximating the entire nonlinear function with one enormous model, each context learns only the best average linear mapping for its own region.

Intuitively, each context answers the question:

> "When representations look like this, what linear transformation has historically produced the best result?"

Composition is the important difference

This viewpoint differs from classical piecewise linear approximation.

Traditional engineering approaches typically work as follows:

- Partition the input space.
- Select one local model.
- Produce the output.

Deep networks are fundamentally compositional.

Every layer performs its own conditional approximation.

The output of one layer becomes the input to the next.

Consequently,

- Layer 1 learns local transformations of raw inputs.
- Layer 2 learns local transformations of Layer 1 representations.
- Layer 3 learns local transformations of Layer 2 representations.
- Every layer operates in its own learned representation space.

The network therefore becomes a hierarchy of locally optimal conditional transformations.

Knowledge as transformations rather than objects

Classical associative memories retrieve stored objects.

This viewpoint suggests that deep networks retrieve stored transformations.

Each fractional associative memory contributes something like:

> "When representations enter this region, this transformation has usually been successful."

The knowledge stored by the network is therefore procedural.

It describes how representations should evolve.

The role of context

A context signal determines which fractional memories participate.

Only a subset of the available parameters contributes to any particular computation.

Each active subset therefore specializes through repeated exposure to similar examples.

Learning gradually sculpts each memory according to the statistics of the contexts that activate it.

The manifold viewpoint

Suppose the data manifold is covered by many overlapping neighbourhoods.

Each neighbourhood accumulates experience about its local statistics.

Rather than storing isolated examples, each fractional memory gradually learns:

- local correlations,
- local trends,
- local transformations,
- local regularities.

Generalization then occurs because nearby inputs often visit similar contexts and therefore benefit from similar learned statistical models.

Out-of-distribution behaviour

This viewpoint also offers an intuitive explanation for out-of-distribution inputs.

An unfamiliar input may activate:

- contexts that received little training,
- unusual combinations of fractional memories,
- local statistical models outside the regions where they were learned.

The network still produces an output.

However, the composed local transformations may no longer correspond to any meaningful region of the training manifold.

Capacity and entropy

The usefulness of a fractional associative memory depends on more than parameter count.

It also depends on:

- the entropy of the routing signal,

- how finely contexts partition the manifold,
- how much overlap exists between neighbouring contexts,
- how much interference occurs between contexts,
- how effectively local statistical models can be reused.

Capacity therefore becomes a property of both storage and organization.

An intuitive picture

Imagine an atlas used in geography.

No single page describes the entire world.

Instead,

- every page describes one region,
- neighbouring pages overlap,
- together they describe a complete surface.

A compositional neural network may operate similarly.

Each fractional associative memory describes one small region of representation space.

Each layer contributes another collection of local statistical models.

The complete computation emerges by composing these local models across many layers.

No individual memory contains the complete solution.

The intelligence of the network lies in the coordinated interaction of many specialized local transformations.

Towards a mathematical treatment

This viewpoint naturally suggests a mathematical interpretation.

Rather than viewing each context-specific matrix as simply another parameter block, it can be interpreted as estimating the best average conditional linear mapping over the region of representation space associated with that context.

The network then becomes a composition of locally optimized conditional linear operators.

Whether this perspective leads to useful theoretical results remains an open question. Nevertheless, it offers a different way to think about what is stored in a deep network. Instead of viewing the parameters as memorized examples or feature detectors alone, they can be viewed as a distributed

collection of local statistical transformation rules whose composition produces the network's overall behaviour.